JTFp FOM Development Process

- Details of scenario and modeling responsibilities were defined.
- Derived a new taxonomy for FOM that is self consistent and sensible for all federation members.
- Each federate built FOMette to support their responsibilities as defined by scenario timeline.

JTFp FOM Development Tools

- OMT Tables - Microsoft Excel
- Paradigm Plus (supplemental)
- Communication
  - » E-mail
  - » JTFp FTP site
  - » JTFp homepage on WWW.
  - » Video Teleconference

JTFp FOM Characteristics

- Mix of discrete and aggregrate level player classes in scenario representation domain.
- Explicit environment class structure and associated interactions.
- Contains objects specific to establishing federation control and monitoring.

JTFp FOM Characteristics (continued)

- Class structure has seven (7) top-level (superclass) nodes, and is no more than three (3) levels deep.
- The most significant information is captured in the class structure, interaction and attribute tables - minimal information in association or composition.

JTFp FOM Characteristics (continued)

- Multiple inheritance heavily used to reduce duplication of attributes.

Lessons Learned

- Inadequacies in OMT documentation concerning relationship between SOMs and FOM were significant hurdle early in FOM development.
  - » Resolved specific issues by developing a new taxonomy for FOM.
  - » Provided input to OMT WG to impact OMT documentation.

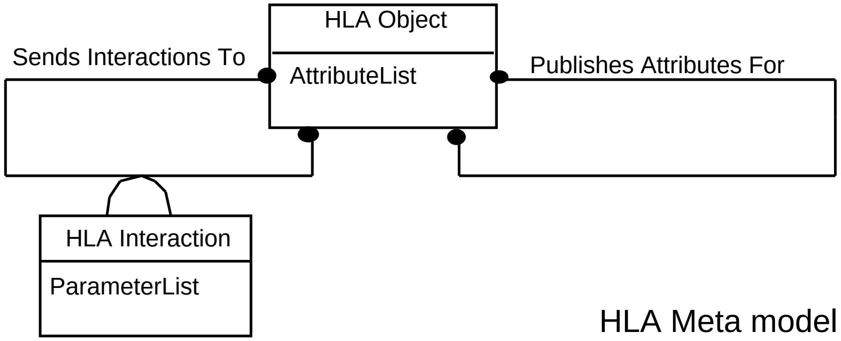
Lessons Learned (continued)

- FOM design requires low-level technical support from federates.
  - » Lack of definition of class methods in SOM make interaction parameters difficult to define.
  - » Resolved by face-to-face meetings and team breakouts.
- Useful to capture interactions that are implemented via publish/subscribe attributes.
  - » Resolved by adding "Logical Interaction Table."

Lessons Learned (continued)

- Scenario timeline built using scenario domain objects, their interactions, and mappings to responsible federates was a pivotal aid in FOM design process.

Logical Interactions



HLA Meta model

- Logical interactions are implemented in HLA by either publishing attributes or sending interactions - a key design decision for FOM development.
- HLA interaction parameters can consist of any data accessible by the sending federate.
- Published class attributes may be involved in more than one logical interaction - a design decision not captured by current HLA OMT tables.